```
Algorithm LEFT-EDGE (N, I)
begin
   FORM-INTERVAL(N, I);
   FORM-HCG(I,HCG);
   d = DENSITY(HCG);
   let T = {T1 T2, ..., Td} denote the set of routing
      tracks from top to bottom;
   SORT-INTERVAL(I);
   for i = 1 to n do
      for j = 1 to d do
         if DOES-NOT-OVERLAP(Ii, Tj) then
            assign interval Ii to Tj;
   for i = 1 to n do
      (* connect the vertical segments of net Ni to its *)
      (* horizontal segment *)
      VERTICAL-SEGMENT(left(Ii), left(Ni));
      VERTICAL-SEGMENT(right(Ii), right(Ni));
end.
```

Figure 9.26: Algorithm LEFT-EDGE

assigned to track 1 since it does not intersect with N_1. The following theorem which establishes the optimality of LEA is easy to prove.

Theorem 14 *Given a two-layer channel routing problem with no vertical constraints, LEA produces a routing solution with minimum number of tracks.*

The input to the algorithm is a set of two-terminal nets $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$. Procedure FORM-INTERVAL forms interval set $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ from $\mathcal{N}$. Once the intervals are formed, FORM-HCG forms the horizontal constraint graph HCG from $\mathcal{I}$. Note that the HCG is a interval graph corresponding to interval set $\mathcal{I}$. Procedure DENSITY computes the maximum clique size in HCG. This maximum clique size is a lower bound on the given channel routing problem instance. SORT-INTERVAL sorts the intervals in $\mathcal{I}$ in the ascending order of their x-coordinate on their left edge. Procedure VERTICAL-SEGMENT connects the vertical segments with the corresponding horizontal segment. The time complexity of this algorithm is $O(n \log n)$, which is the time needed for sorting n intervals.

The assumption that no two nets share a common end point is too restrictive, and as a result LEA is not a practical router for most channel routing problems. The restrictions placed on the router in order to achieve optimal results are not practical for most channel routing problems. However, LEA can be used to route PCB routing problems with vertical constraints since there is sufficient space between the adjacent pins to create a *jog*. LEA is also useful as a initial router for routing of channels with vertical constraints. The basic

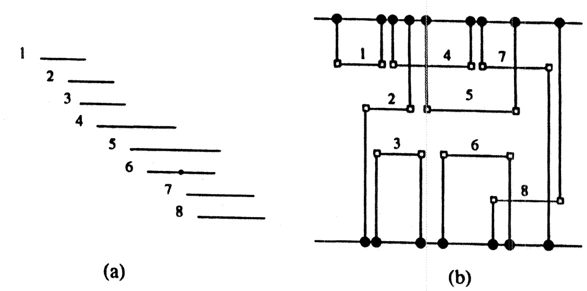

Figure 9.27: Left-edge channel routing.

idea is to create a layout with design rule violations and then use clean up procedures to remove the violations.

9.4.2.2 Dogleg Router

One of the drawbacks of LEA is that it places an entire net on a single track. It has been observed that this leads to routings with more tracks than necessary. Consider Figure 9.28(a), which shows a simple channel routing problem that has been routed using LEA and uses three tracks. On the other hand, if a dogleg is introduced in net N_2, the same problem can be routed using only two tracks. We recall that a dogleg is a vertical segment that is used to maintain the connectivity of two trunks (subnets) that are on two different tracks. The insertion of doglegs, may not necessarily reduce the channel density. A badly placed dogleg can lead to an increase in channel density. Finding the smallest number and locations of doglegs to minimize the channel density is shown to be NP-complete [Szy85].

Deutsch [Deu76] proposed an algorithm known as *dogleg router* by observing that the use of doglegs can reduce channel density. The dogleg router is that it allows multi-terminal nets and vertical constraints. Multi-terminal nets may have terminals on both sides of the channel and often form long horizontal constraint chains. In addition, there are several critical nets, such as clock nets, which pose problems because of their length and number of terminals. These type of nets can be broken into a series of two-terminal subnets using doglegs and each subnet can be routed on a different track. Like LEA, the Dogleg router uses a reserved layer model. Restricting the doglegs to the terminal positions reduces the number of unnecessary doglegs and consequently reduces the number of vias and the capacitance of the nets. The dogleg router cannot handle cyclic vertical constraints.

The dogleg router introduces two new parameters: *range* and *routing sequence*. Range is used to determine the number of consecutive two-terminal

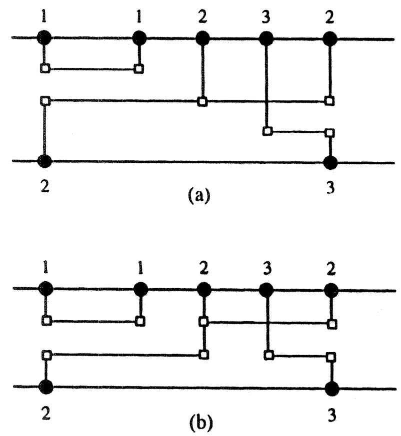

Figure 9.28: Using a dogleg to reduce channel density.

subnets of the same net which can be placed on the same track. Increasing the range parameter will result in fewer doglegs. The routing sequence specifies the starting position and the direction of routing along the channel. Unlike LEA, the routing can start from any end and work towards the opposite end. Different results can be obtained by starting at different corners: top-left, top-right, bottom-left, bottom-right. Furthermore, instead of starting from the top to the bottom or from the bottom to the top, the algorithm can alternate between topmost and bottommost tracks. This scheme results in eight different routing sequences: top-left → bottom-left, top-left → bottom-right, top-right → bottom-left, top-right → bottom-right, bottom-left → top-left, bottom-left → top-right, bottom-right → top-left, and bottom-right → top-right. (The left side of the arrow indicates the starting corner and the right side of the arrow indicates the alternate corner). Consider the example shown in Figure 9.29(a). If the range is set to 1 and we set the routing sequence to top-left → bottom-right, then Figure 9.29(b) shows routing steps in dogleg router. Notice that nets N_2 and N_3 use doglegs.

The complexity of the algorithm is dominated by the complexity of LEA. As a result, the complexity of the algorithm is $O(n \log n + nd)$, where n is the total number of two-terminal-nets after decomposition and d is the total number of tracks used. Note that the parameter's range and routing sequence can be changed to get different solutions of the same routing problem. A large value of range keeps the number of doglegs smaller. If the number of two-terminal subnets of a net is less than the value of a range, then that net is routed without any dogleg. Varying the routing sequence can also lead to a reduced

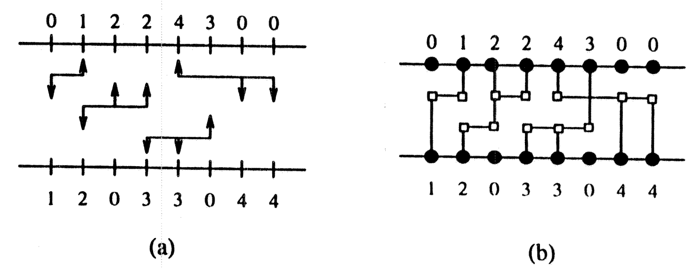

Figure 9.29: Example routed by dogleg router.

channel height. Dogleg router can easily be extended to gridless routing model. Experimentally dogleg routers achieve far superior results as compared to LEA, often requiring very few tracks beyond the channel density.

9.4.2.3 Symbolic Channel Router: YACR2

LEA does not allow vertical constraints thereby making it impractical for most of the channel routing problems. If a given channel is routed using LEA, then vertical constraint violations may be introduced by the router which need to be removed to get a legal routing solution. Note that a vertical constraint violation is a localized problem and may be resolved by anyone of the two methods:

1. local rip-up and reroute

2. localized maze routing.

In the second approach, vacant space surrounding the column in which vertical constraint violation occurred can be used to resolve the violation. Usually several horizontal segments of tracks as well as several vertical columns are not used for routing of any nets. Since the general maze routing technique is very time consuming and vertical constraint violations are local in nature, special maze routing techniques can used to remove vertical constraint violations. In case any vertical constraint violations cannot be resolved, new tracks can be added to resolve the constraints.

Based on these observations, Reed, Sangiovanni-Vincentalli, and Santamauro [RSVS85] proposed YACR2 (Yet Another Channel Router). In order to explain how vertical constraint violations are handled in YACR2 we define the concept of vertical overlap factor, which indicates the total number of tracks that a vertical constraint violation spans. Precisely stated, let us assume that column c_i has a vertical constraint violation between net N_t that has to be connected to the top boundary and net N_b that has to be connected